Run Your Own FRED MCP Server

Nov 4, 2026

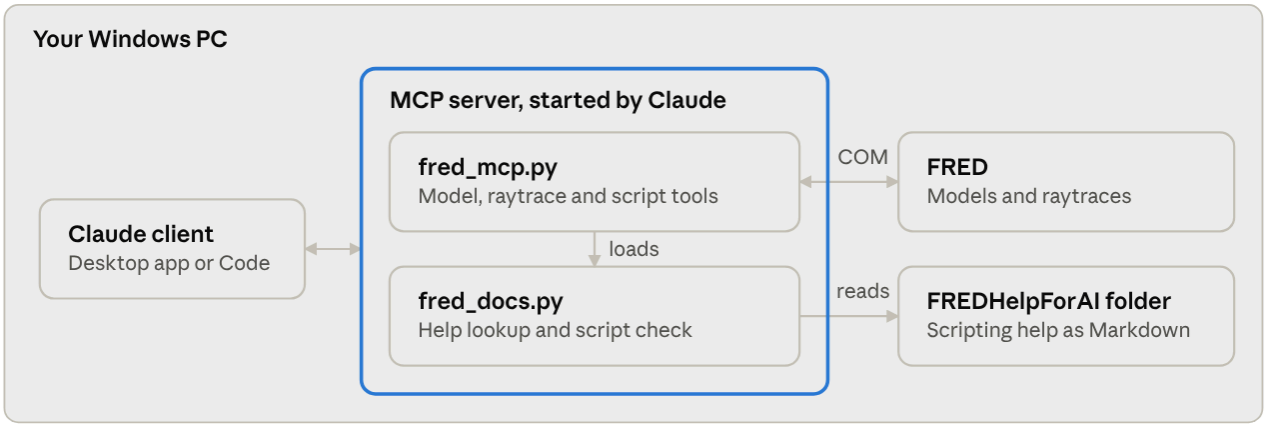
What this is

This package lets Claude look up FRED's scripting help, check scripts against it, and drive your own copy of FRED on your own PC. Claude can open models, run raytraces and run FRED BASIC scripts, then read the results back. Everything runs locally through FRED's COM automation interface.

Item	What it is	What it needs
`FRED_MCP\fred_mcp.py`	The MCP server: model, raytrace and script tools	Python, FRED
`FRED_MCP\fred_docs.py`	Help lookup and script checking tools, loaded by `fred_mcp.py`	Python
`FREDHelpForAI\`	FRED's scripting help as Markdown: 46 command groups, 99 data structures and 33 guides	Read only

The server reads `FREDHelpForAI` but never writes to it. The help tools work without FRED running. Only the model, raytrace, `call_fred` and `run_script` tools need FRED.

Claude reaches FRED and its help through one local server



How the parts connect

The Claude client starts the server for you, and the server keeps one connection open to FRED.

Requirements

You need a Windows PC with FRED, Python and a Claude client all on the same machine, because the server talks to FRED through Windows COM.

Item	Requirement	Notes
Windows	Required	COM is Windows only
FRED	Installed and licensed	The help describes a recent release, so newer commands may be missing from older versions
Python	3.11 or newer	Developed on 3.12. Anaconda or python.org both work (SDK install notes)
Claude client	Claude desktop app or Claude Code	Either one starts the server for you. Other MCP clients are covered below
Node.js	Optional	Only needed for the `mcp dev` test tool

Installation

Setup takes five steps and about ten minutes. Replace `YOUR-NAME` with your Windows user name throughout.

1. Unpack the package

Unzip it so the two folders sit side by side. Any location works, but you will need the full path later.

```
C:\Users\YOUR-NAME\FRED-MCP\
    FRED_MCP\          fred_mcp.py, fred_docs.py, README.md
    FREDHelpForAI\     commands\, guides\, data-structures.md, INDEX.md
```

Keep `fred_mcp.py` and `fred_docs.py` in the same folder. The server needs both.

2. Find your Python

Open PowerShell, or the Anaconda Prompt if you use Anaconda, and run:

```
python -c "import sys; print(sys.executable, sys.version)"
```

Note the path it prints, such as `C:\Users\YOUR-NAME\anaconda3\python.exe`. Check that the version is 3.11 or newer.

3. Install the Python packages

Installing through `python -m pip` puts the packages into the Python you just found.

```
python -m pip install "mcp[cli]" pywin32
python -c "import mcp, win32com.client; print('OK')"
```

The second command should print `OK`.

4. Test the server by hand

This catches most problems before a Claude client is involved.

```
$env:FRED_DOCS_DIR = "C:\Users\YOUR-NAME\FRED-MCP\FREDHelpForAI"
python "C:\Users\YOUR-NAME\FRED-MCP\FRED_MCP\fred_mcp.py"
```

The command should sit silently, waiting for a client. That means it worked, so press Ctrl+C to stop it. An error message here means step 3 needs fixing.

5. Connect a Claude client

Use one of these two options, or see Using another MCP client below.

Claude desktop app. Open Settings, choose Developer, then click Edit Config. Use that button rather than browsing for the file, because its location can vary between PCs. Add this, with your own paths:

```
{
  "mcpServers": {
    "fred": {
      "command": "C:\\Users\\YOUR-NAME\\anaconda3\\python.exe",
      "args": ["C:\\Users\\YOUR-NAME\\FRED-MCP\\FRED_MCP\\fred_mcp.py"],
      "env": {
        "FRED_DOCS_DIR": "C:\\Users\\YOUR-NAME\\FRED-MCP\\FREDHelpForAI"
      }
    }
  }
}
```

- Every backslash must be doubled in this file.
- Use the full Python path from step 2. The app may not find a plain `python`.
- If the file already has an `mcpServers` section, add the `fred` entry inside it.
- `FRED_DOCS_DIR` sets the help folder. It's only needed if that folder doesn't sit beside FRED_MCP.

Save the file, then fully quit Claude from the system tray and reopen it. Closing the window is not enough (config details).

Claude Code. Run this in one line, keeping the server name before `-e`:

```
claude mcp add fred -e FRED_DOCS_DIR="C:\Users\YOUR-NAME\FRED-MCP\FREDHelpForAI" -- "C:\Users\YOUR-NAME\anaconda3\python.exe" "C:\Users\YOUR-NAME\FRED-MCP\FRED_MCP\fred_mcp.py"
```

Then run `claude mcp list` and check that `fred` is connected.

Check it works

Try these four prompts in a new chat, in order. The first two prove the help works without FRED, and the last two prove the link to FRED. Claude may ask your permission the first time it uses each tool.

1. **"Use the FRED tools to search the help for gaussian beam source."** Claude should list commands such as `SetSourceDirAstigmaticGausBeam`.
2. **"Check this FRED script for errors: Sub Main() / TraceCreatt / End Sub."** Claude should flag `TraceCreatt` and suggest `TraceCreate`.
3. **"Use model_info to check the FRED connection."** FRED should appear on screen if it wasn't already open. The reply says either "started a new FRED instance" or "connected to an already-running FRED".
4. **"Create a new FRED model and tell me what's in it."** A new document should open in FRED.

If all four work, your setup is complete.

Using another MCP client

The server and help files aren't tied to Claude, so any MCP client that can start a local server on the same PC should work. So far it has only been used with Claude. Follow steps 1 to 4, then add the server to your client with these values.

Setting	Value
Server type	Local (stdio)
Name	`fred`
Command	Full path to `python.exe`, from step 2
Arguments	Full path to `fred_mcp.py`

Setting	Value
Environment variable	`FRED_DOCS_DIR`, set to the full path of `FREDHelpForAI`

Each client has its own config format, so check its MCP documentation. The test prompts above work in any client, though the tool names and approval steps may look different.

- **Results depend on the model.** The server asks clients to look commands up rather than guess, but not every client passes that on. If a model invents commands, tell it to use `fred_doc_search` and `fred_doc_lookup` first.
- **Check the approval settings.** Make sure your client asks before running `run_script` and `call_fred`.
- **Check where your data goes.** Help text, scripts and model details returned by the tools go to your AI provider.

AI tools that can't start a local server can't drive FRED, but they can still use the help. Add the Markdown files from `FREDHelpForAI` as documents or a knowledge base, starting with `INDEX.md`. The set is about 4 MB across 80 or so files, so check your tool's limits.

Tools and settings

The server gives Claude 12 tools. You don't call them yourself: describe the task and Claude picks the tool.

Tool	What it does	Needs FRED
`fred_doc_search`	Keyword search across commands, structures and guides	No
`fred_doc_lookup`	Full help entry for one command or `T_` structure	No
`fred_doc_list`	Lists usage groups, or the commands in one group	No
`fred_doc_guide`	Reads a guide or worked example	No
`fred_check_script`	Flags unknown command names and shows the syntax of the rest	No
`open_model`	Opens a `.frx` or `.frd` file	Yes
`new_model`	Creates an empty document	Yes
`model_info`	Reports the connection, active document and open-document count	Yes

Tool	What it does	Needs FRED
`save_model_as`	Saves the active document, refusing to overwrite unless told to	Yes
`raytrace`	Runs `TraceCreate`, or `TraceCreateDraw` to draw rays as well	Yes
`call_fred`	Calls one documented command. Not for `T_` structures	Yes
`run_script`	Compiles a FRED BASIC script and runs one Sub or Function, returning its value	Yes

The defaults suit most people. To change one, add it to the `env` block of your config.

Setting	Default	Purpose
`FRED_DOCS_DIR`	The folder beside FRED_MCP	Path to the `FREDHelpForAI` folder
`FRED_PROGID`	`FRED.Application`	COM name used to start FRED
`FRED_ATTACH`	`1`	`0` skips the check for a running FRED. FRED still reuses a running copy, so `0` does not force a second one

Using it safely

Treat `run_script` like running a macro from someone else: it executes real code on your PC with whatever access FRED has. Read scripts before approving them when the model or your files matter.

- **Claude works on your open model.** If FRED is already running, the server uses that copy and its active document. Save your work first.
- **Scripts can do more than edit a model.** FRED BASIC can read and write files and start other programs.
- **The script check is limited.** It checks command names only, not arguments or logic. `force=True` skips it, so a script that passes can still be wrong.
- **Quitting Claude can close FRED.** If the server started FRED, FRED may close when the server stops, and open documents may be lost. Save before you quit.
- **Keep approval prompts on for `run_script` and `call_fred`.** These are the tools that can change things.

Troubleshooting

Start with the server's log, which records why it failed to start. In the desktop app it is `mcp-server-fred.log`, in the `logs` folder next to your config file. That folder is usually `%APPDATA%\Claude\logs`.

What you see	Likely cause	Fix
No FRED tools in Claude	Claude wasn't fully restarted, or the config is invalid	Quit from the system tray. Check for single backslashes and missing commas
Server fails to start or `ENOENT` in the log	The app can't find `python`	Use the full path to `python.exe` in `command`
`No module named 'mcp'` or `'win32com'`	Packages went into a different Python	Run step 3 using the full path to the Python in your config
`DLL load failed` in the log, but step 4 worked	Anaconda started outside its prompt	Try a python.org install, or add Anaconda's `Library\bin` folder to `PATH` in the `env` block
`FRED docs folder not found`	`FRED_DOCS_DIR` is wrong or missing	Point it at the folder that contains `commands`
`Invalid class string`	FRED isn't registered for COM on this PC	Check that FRED is installed here, then reinstall it if needed
`No FRED document is open`	Nothing is open in FRED	Ask Claude to open a model or create a new one
A real command is refused as undocumented	Some help headings contain typos, which are listed in `README.md`	Tell Claude the correct name. `run_script` accepts `force=True`
A long raytrace seems to hang	FRED calls run one at a time, and the client may stop waiting	Check progress in FRED. Other FRED tools resume once it finishes

To test the server without an AI client, run `mcp dev` followed by the quoted path to `fred_mcp.py`. This opens the MCP Inspector and needs Node.js. Set `FRED_DOCS_DIR` first as in step 4.

Updating

To update, replace the old files with the new package. The help folder updates without a restart, but the server code needs one.

What changed	What to do	Restart needed
Help files	Replace the contents of `FREDHelpForAI`	No, the index refreshes on the next help lookup
Server code	Replace `fred_mcp.py` and `fred_docs.py`	Yes, fully quit and reopen Claude

Don't keep your own notes or scripts in `FREDHelpForAI`, because a help update replaces that folder. Your config stays valid as long as the paths stay the same.